

CLAIMS

What is claimed is:

1 1. A method for managing a code sequence, comprising:
2 determining first intermediate correlation values for a first plurality of sample sequences
3 during a first clock cycle;
4 determining second intermediate correlation values for the first plurality of sample
5 sequences during a second clock cycle; and
6 determining correlation outputs for the first plurality of sample sequences from the first
7 and second intermediate correlation values.

1 2. The method of Claim 1, wherein determining the first intermediate correlation values
2 comprises processing coefficients in a first code sequence group in parallel with corresponding
3 sample values in corresponding sample sequence groups from the first plurality of sample
4 sequences.

1 3. The method of Claim 1, wherein determining the second intermediate correlation
2 values comprises processing coefficients in a second code sequence group in parallel with
3 corresponding sample values in corresponding sample sequence groups from the first plurality of
4 sample sequences.

1 4. The method of Claim 1, wherein determining correlation outputs for the first plurality
2 of sample sequences comprises taking a sum of the first and second intermediate correlation
3 values for each of the first plurality of sample sequences.

1 5. The method of Claim 1, further comprising:

2 determining first intermediate correlation values for a second plurality of sample values
3 during a third clock;
4 determining second intermediate correlation values for the second plurality of sample
5 values during a fourth clock; and
6 determining correlation output values for the second plurality of sample value from the
7 first and second intermediate correlation values.

1 6. A method for managing a code sequence, comprising:
2 processing a first group of coefficients in the code sequence with a first group of sample
3 values in a received sample to determine a first intermediate correlation value during a first clock
4 cycle;
5 processing a second group of coefficients in the code sequence with a second group of
6 sample values in the received sample to determine a second intermediate correlation value during
7 a second clock cycle; and
8 determining a correlation output from the first and second intermediate correlation
9 values.

1 7. The method of claim 6, wherein the code sequence comprises L coefficient and the
2 first and second group of coefficients in the code sequence each comprises n coefficients.

1 8. The method of claim 7, wherein the first and second group of sample values in the
2 received sample each comprises n sample values.

1 9. The method of claim 6, wherein the first and second group of coefficients in the code
2 sequence are contiguous.

1 10. The method of claim 6, wherein the first and second group of sample values in the
2 received sample are contiguous.

1 11. The method of claim 6, wherein processing the first group of coefficient in the code
2 sequence with the first group of sample values in the received sample comprises determining a
3 sum of the products of the first group of coefficients in the code sequence with the first group of
4 sample values in the received sample.

1 12. The method of claim 6, wherein processing the second group of coefficients in the
2 code sequence with the second group of sample values in the received sample comprises
3 determining a sum of the products of the second group of coefficients in the code sequence with
4 the second group of sample values in the received sample.

1 13. The method of claim 6, wherein determining the correlation output from the first and
2 second intermediate correlation values comprises taking the sum of the first and second
3 intermediate correlation values.

1 14. A method for managing a code sequence, comprising:
2 organizing the code sequence, having a plurality of contiguous coefficients, into a
3 plurality of contiguous code sequence groups;
4 selecting a number of sample sequences to process in parallel where each of the sample
5 sequences has contiguous sample values from a received sample;
6 organizing contiguous sample values from each of a first set of sample sequences to
7 process in parallel into a first set of contiguous sample sequence groups; and
8 processing coefficients in each of the code sequence groups in parallel with
9 corresponding sample values in corresponding sample sequence groups from the first set of

10 sample sequences, where each of code sequence groups is processed during a different clock
11 cycle.

1 15. The method of Claim 14, further comprising:
2 organizing contiguous sample values from each of a second set of sample sequences to
3 process in parallel into a second set of contiguous sample sequence groups; and
4 processing coefficients in each of the code sequence groups in parallel with
5 corresponding sample values in corresponding sample sequence groups from the second set of
6 sample sequences, where each of the code sequence groups is processed during a different clock
7 cycle.

1 16. The method of Claim 14, further comprising:
2 determining a correlation output for each of the sample sequences; and
3 determining a synchronization point for the code sequence from the correlation outputs.

1 17. The method of Claim 16, wherein determining a synchronization output comprises
2 determining a correlation output having a highest numerical value.

1 18. The method of Claim 14, wherein a first sample value in a first sample sequence
2 includes a first sample value in the received sample and each consecutive sample sequence
3 includes a next contiguous sample value in the received sample as a first sample value the
4 consecutive sample sequence.

1 19. The method of Claim 14, wherein processing comprises determining a sum of the
2 products of the coefficients in each of the code sequence groups with each of the sample values in
3 corresponding sample sequence groups from the first set of sample sequences.

1 20. The method of Claim 14, wherein the code sequence comprises a plurality of L
2 contiguous values.

1 21. The method of Claim 20, wherein the code sequence is organized into a plurality of n
2 code sequence groups.

1 22. The method of Claim 21, wherein a number, d, sample sequences are selected to
2 process in parallel where each of the sample sequences has L contiguous sample values from the
3 sample.

1 23. The method of Claim 22, wherein the first set of sample sequences is organized into
2 a plurality of contiguous sample sequence groups having n values each.

1 24. The method of Claim 14, wherein the code sequence is organized into L/n groups.

1 25. The method of Claim 14, wherein the processing is completed after L/n clocks.

1 26. A method for managing a code sequence, comprising:
2 organizing the code sequence, having L contiguous coefficients, into a number of
3 contiguous code sequence groups having n coefficients each;
4 selecting a number of sample sequences, d, to process in parallel where each of the
5 sample sequences has L contiguous sample values from a received sample, where a first sample
6 value in a first sample sequence is a first sample value in the received sample and each
7 consecutive sample sequence includes a next contiguous sample value in the received sample as a
8 first sample value in the consecutive sample sequence;

9 organizing sample values from each of a first set of d sample sequences into a first set of
10 sample sequence groups having n values each; and
11 processing coefficients in each of the code sequence groups in parallel with
12 corresponding sample values in corresponding sample sequence groups from the first set of d
13 sample sequences, where each of the code sequence groups is processed during a different clock
14 cycle.

1 27. The method of Claim 26, further comprising:
2 organizing sample values from each of a second set of d sample sequences into a second
3 set of contiguous sample sequence groups having n values each; and processing values in each of
4 the code sequence groups in parallel with corresponding sample values in corresponding sample
5 sequence groups from the second set of d sample sequences, where each of the code sequence
6 groups is processed during a different clock cycle.

- 1 28. The method of Claim 26, further comprising:
 - 2 determining a correlation output for each of the sample sequences; and
 - 3 determining a synchronization point for the code sequence from the correlation outputs.

1 29. The method of Claim 28, wherein determining a synchronization output comprises
2 determining a correlation output having a highest numerical value.

1 30. The method of Claim 26, wherein the code sequence is organized into L/n groups.

1 31. The method of Claim 26, wherein processing comprises determining a sum of the
2 products of the coefficients in each of the code sequence groups with each of the sample values in
3 corresponding sample sequence groups from the first set of d sample sequences.

1 32. The method of Claim 26, wherein the processing is completed after L/n clocks.

1 33. A correlator unit, comprising:

2 a plurality of code sequence registers that store coefficients from a code sequence group,

3 the plurality of code sequence registers storing coefficients from one code sequence group of a

4 plurality of code sequence groups at a time;

5 a plurality of sample registers that store sample values from a plurality of sample

6 sequences that are processed in parallel; and

7 a processing unit that processes coefficients in each of the plurality of code sequence

8 groups in the plurality of code sequence registers in parallel with corresponding sample values in

9 corresponding sample sequence groups from a first plurality of sample sequences in the plurality

10 of sample registers, where each of the code sequence groups is processed during a different clock

11 cycle.

1 34. The correlator unit of Claim 33, further comprising a plurality of accumulation sub-

2 units each accumulation sub-unit receiving results from the processing unit for a designated

3 sample sequence, each accumulation unit generating a correlation value for the designated sample

4 sequence after each of the code sequence groups are processed.

1 35. The correlator unit of Claim 33, wherein the processing unit processes the

2 coefficients in each of the plurality of the plurality of code sequence groups in the plurality of

3 code sequence registers in parallel with corresponding sample values in corresponding sample

4 sequence groups from a second plurality of sample sequences in the plurality of sample registers,

5 where each of the code sequence groups is processed during a different clock cycle.

1 36. The correlator unit of Claim 34, further comprising correlation output processor
2 that determines a synchronization point for the code sequence from the correlation outputs.

1 37. The correlator unit of Claim 36, wherein the correlation output processor determines
2 a synchronization point from a correlation output having a highest numerical value.

1 38. The correlator unit of Claim 33, wherein the processing unit determines a sum of
2 products of the coefficients in each of the code sequence groups with corresponding sample
3 values in corresponding

1 39. A correlator unit, comprising:
2 a plurality of n code sequence registers that store n coefficients from a code sequence
3 group, the plurality of n code sequence registers storing coefficients from one code sequence
4 group of a plurality of code sequence groups at a time;
5 a plurality of n+d-1 sample registers that store sample values from a plurality of d sample
6 sequences that are processed in parallel; and
7 a processing unit that processes coefficients in each of the plurality of code sequence
8 groups in the plurality of n code sequence registers in parallel with corresponding sample values
9 in corresponding sample sequence groups from a first plurality of d sample sequences in the
10 plurality of n+d-1 sample registers, where each of the code sequence groups is processed during a
11 different clock cycle.

1 40. The correlator unit of Claim 39, further comprising an accumulation sub-unit,
2 corresponding to each of the d sample sequences that are processed in parallel, that receives
3 results from the processing unit for a designated sample sequence and that determines a

4 correlation output for the designated sample sequence after each of the code sequence groups are
5 processed.

1 41. The correlator unit of Claim 39, wherein the processing unit processes the
2 coefficients in each of the plurality code sequence groups in the plurality of n code sequence
3 registers in parallel with corresponding sample values in corresponding sample sequence groups
4 from a second plurality of d sample sequences in the plurality of $n+d-1$ sample registers, where
5 each of the code sequence groups is processed during a different clock cycle.

1 42. The correlator unit of Claim 40, further comprising correlation output processor
2 that determines a synchronization point for the code sequence from the correlation outputs.

1 43. The correlator unit of Claim 42, wherein the correlation output processor determines
2 a synchronization point from a correlation output having a highest numerical value.

1 44. The correlator unit of Claim 39, wherein the processing unit determines a sum of
2 products of the coefficients in each of the code sequence groups with each of the sample values in
3 corresponding sample sequence groups from the first set of d correlation sequences.

1 45. The correlator unit of Claim 39, wherein the processing is completed after L/n
2 clocks.

1 46. A correlator unit, comprising:
2 means for storing coefficients from a code sequence group, the means for storing
3 coefficients storing coefficients from one code sequence group of a plurality of code sequence
4 groups at a time;

5 means for storing sample values from a plurality of sample sequences that are processed
6 in parallel; and

7 means for processing coefficients in each of the plurality of code sequence groups in the
8 means for storing coefficients in parallel with corresponding sample values in corresponding
9 sample sequence groups from a first plurality of sample sequences in the means for storing
10 sample values, where each of the code sequence groups is processed during a different clock
11 cycle.